Хоанг Ван Кует, аспирант кафедры оптимизации систем управления Института кибернетики ТПУ.
E-mail: student8050@sibmail.com
Область научных интересов: исследование методов поддержки работы с семантическими базами знаний и их эффективности.

Тузовский Анатолий Федорович, д-р техн. наук, профессор кафедры оптимизации систем управления Института кибернетики ТПУ.
E-mail: tuzovskyaf@tpu.ru
Область научных интересов: семантические технологии, онтологии, интернет технологии, технологии разработки программ.

УДК 004.415

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ СЕМАНТИЧЕСКИХ БАЗ ДАННЫХ

Хоанг Ван Кует, А.Ф. Тузовский

Томский политехнический университет
E-mail: student8050@sibmail.com

Предлагается метод контроля доступа пользователей к семантическим базам данных. Выделены две задачи контроля доступа: контроль прямого доступа к данным и контроль логических выводов на получаемых данных. Рассмотрены основные свойства отношений между объектами в семантических данных. Предложен метод для контроля логических выводов несанкционированных пользователей.

Ключевые слова
Семантическая база данных, безопасность доступа, триплеты, логический вывод, несанкционированный доступ.

В настоящее время значительно повышаются возможности взаимодействия между источниками данных, увеличивается количество баз данных, упрощаются способы работы пользователей с ними. В связи с этим очень важной становится проблема защиты данных от несанкционированного использования. Повышается актуальность задач контроля доступа к данным и их использования с учётом заданных прав пользователей.

В то же время в последние годы очень активно развивается такое новое направление информатики, как семантические технологии, которые начинают играть важную роль в разработке информационных систем организаций. Применяемые в этих технологиях семантические данные отличаются от реляционных данных тем, что на их основе может выполняться логический вывод, позволяющий получать новую информацию. Для обеспечения безопасности работы с такими данными предложены разные подходы [1], но они не предоставляют возможности контроля доступа к частям данных и проверки возможности выполнения пользователями на доступных данных нежелательных логических выводов. Для безопасности семантических данных необходимо решить следующие задачи: управление прямого доступа к данным и управление возможными логическими выводами.

В данной статье для решения перечисленных выше задач предложены метод обеспечения доступа пользователей к данным на основе использования меток безопасности триплетов RDF-данных и метод контроля логических выводов пользователей с помощью алгоритма обнаружения их нарушений.

Метка безопасности для триплета RDF-данных

Каждый RDF-документ состоит из графа, являющегося набором триплетов, состоящих из троек: субъект (r), предикат (x), объект (v) [2]. В целях безопасности RDF-данных каждый элемент триплета имеет разные права доступа, т. е. они находятся на разных уровнях безопасности согласно своим меткам (sl_1, sl_2, …, sl_n), созданным пользователями. Например, sl_1 соответствует открытым данным, sl_2 соответствует конфиденциальным данным, …, sl_n соответствует сверхсекретным данным.

В каждый из компонентов триплете (субъект, предикат, объект) может иметь одинаковую метку *sl* или разные метки безопасности (sl_r, sl_x, sl_v) (где sl_r – метка для субъекта, sl_x – метка для предиката, sl_v – метка для объекта). В том случае, когда три элемента имеют одинаковую метку *sl*, то *sl* является меткой безопасности для триплета ($sl = sl_{tr}$, где sl_{tr} – метка для триплета). Если три метки являются разными ($sl_r \neq sl_x \neq sl_v$), то надо образовать наименьшую верхнюю границу чувствительности sl_{max} этих меток, тогда sl_{max} является меткой безопасности триплета $\mathrm{sl_{tr}}$ ($sl_{tr} = sl_{max}$).

Один и тот же ресурс может являться субъектом или объектом (иногда предикатом) в разных ситуациях, следовательно, он может иметь разные метки безопасности (sl_1, sl_2, …, sl_n) в зависимости от своего положения. Триплет, обладающий ресурсом в конкретной роли (субъект или объект), имеет конкретную метку sl_t, охватывающую соответствующую метку обеспечения ресурса в данной позиции, и $sl_t \geq \max(sl_1, sl_2, \ldots, sl_n)$.

Модель контроля доступа пользователей

Предлагаемая модель построена на основе мандатной политике безопасности, основанной на мандатном разграничении доступа, которая определяется четырьмя условиями [3]:

- все пользователи и триплеты базы данных однозначно идентифицированы;
- задана решётка уровней безопасности информации;
- каждому триплету базы данных присвоен уровень безопасности, определяющий ценность содержащейся в нем информации;
- каждому пользователю присвоен уровень доступа, определяющий уровень доверия к нему в семантической базе данных.

В семантической базе данных каждый триплет имеет свой уровень безопасности (*AC*), указатель которого может иметь значения из множества меток безопасности $L = \{$неклассифицированный (L_U), конфиденциальный (L_C), секретный (L_S) и сверхсекретный (L_{TS})$\}$, где $L_U < L_C < L_S < L_{TS}$. Каждый пользователь имеет уровень доступа $\mathrm{AC_s}$, включающий права выполнения таких операций с данными, как $R = \{$*read*, *write*, *append*, *execute*$\}$, где *read* – право на чтение триплета, *write* – право на запись триплета, *append* – право на запись в конец объекта, *execute* – право на добавление или удаление триплета.

В соответствии с мандатной моделью безопасность базы данных определяется следующими свойствами:

- *Свойство простой безопасности («простое» свойство безопасности)*: Пользователь может иметь право доступа на чтение триплета только в случае, когда уровень доступа пользователя не ниже уровня безопасности триплета.
- *Свойство «звезда» безопасности (свойство «звезда»)*: Пользователь может иметь доступ к триплету в случае, когда уровень безопасности триплета не ниже его уровня доступа. Пользователь может иметь право доступа на запись триплета только в случае, когда его уровень доступа равен уровню безопасности триплета. Он может иметь право доступа на чтение триплета только в случае, когда его уровень доступа не ниже уровня безопасности триплета.
- *Свойство дискреционной безопасности (Свойство безопасности [ds - свойство])*: Каждый доступ содержит, по крайней мере, один элемент из множества прав *R* пользователя на триплет.
- *Невозможность полного общения с бездействующим триплетом*: Пользователь не может читать бездействующий триплет.

Для семантических баз данных предлагаются следующие характеристики поддержки безопасности работы с ними:

- *Свойство чтения*: Пользователь может читать триплет только в том случае, когда его уровень доступа не ниже уровня безопасности триплета.
- *Свойство записи*: Только владелец имеет права на запись (модификацию) своих данных.
- *Безопасность триплета:* Уровень безопасности триплета должен быть не ниже уровня доступа его владельца.
- *Безопасность элементов триплета***:** Уровень безопасности триплета должен быть не ниже уровня безопасности его субъекта, объекта и предиката.

- *Доступ на модификацию триплетов*: Только администратор безопасности базы данных и владелец триплетов имеют права на модификацию уровня безопасности триплетов и уровней доступа пользователей.

Контроль прямого доступа пользователей к базе данных

Под контролем прямого доступа пользователей понимается возможность получать ими данные в соответствии с заданными для них правами на использование данных [4].

В семантических базах данных хранятся разные категории информации, такие как: экономическая, финансовая и т. п. Если уровень безопасности информации AC содержит только один указатель с меткой безопасности из множества L, то пользователь, имеющий уровень доступа типа $AC_s = L_C$ в категории M, может иметь доступ в категории N к данным, имеющим уровень безопасности $AC = L_C$. В данном случае количество возможных вариантов создания уровней безопасности информации в базе данных будет сильно ограниченным. Для решения данной проблемы в уровни безопасности данных могут быть добавлены другие указатели с метками безопасности из множества L.

В данной работе предлагается, что метка безопасности RDF-триплета $AC = (S, P, PS, C)$ (уровень безопасности триплета) включает следующие 4 показателя:

- ***чувствительность*** S – определяет уровень значимости или важности связи (предиката), а также её уязвимости перед несанкционированным лицом;
- ***приватность*** P – определяет права владельца на возможность передачи данной информации другим пользователям;
- ***персональная безопасность*** PS – определяет уровень защиты персональной информации человека или организации;
- ***конфиденциальность*** C – определяет возможность совместного использования данной информации с другими ресурсами.

Каждый показатель может принимать значения из множества L (в данной работе предлагается, что эти показатели могут принимать значение 0 (unclassified), 1 (confidential)).

Право доступа пользователей к данным $AC_s = (S_s, P_s, PS_s, C_s)$ так же включает 4 показателя: чувствительность S, приватность P, персональная безопасность PS и конфиденциальность C. Для контроля доступа пользователя к RDF-триплету необходимо сравнивать уровень права доступа пользователей $AC_s = (S_s, P_s, PS_s, C_s)$ с уровнем безопасности триплета $AC = (S, P, PS, C)$.

Считается, что $AC_s = (S_s, P_s, PS_s, C_s)$ не ниже уровня $AC = (S, P, PS, C)$ только в случае, когда $S_s \geq S$, $P_s \geq P$, $PS_s \geq PS$, $C_s \geq C$. Если $AC_s \geq AC$, то пользователи могут получить доступ к этим данным. Если $AC_s < AC$, то пользователи не имеют права доступа к этим данным, и данные являются невидимыми для пользователей. На рис. 1 показан алгоритм контроля прямого доступа к триплету RDF-данных (алгоритм 1).

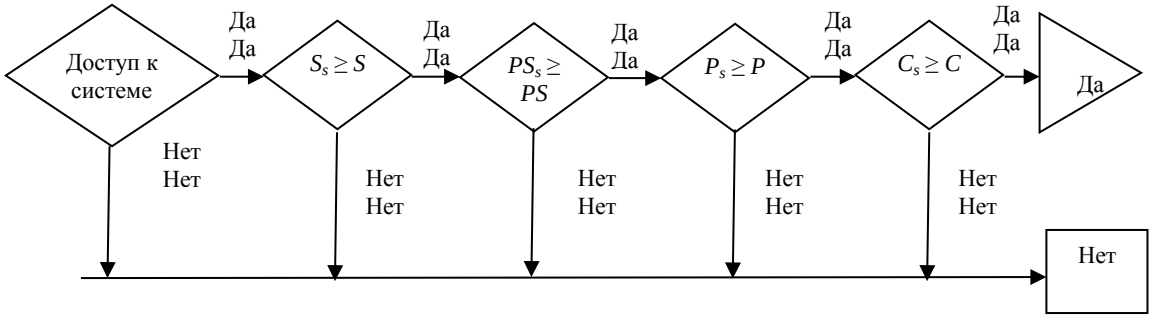

Рис. 1. Алгоритм контроля прямого доступа пользователей к базе данных

В соответствии с этим алгоритмом проверка прав доступа пользователя осуществляется на каждом триплете. Входными данными алгоритма являются: уровень права доступа пользователя и множество индексов чувствительности триплетов RDF-данных. Выходными данными

является множество всех триплетов, к которым пользователь имеет право доступа. На каждом шаге приходится процесс сравнения каждого критерия индекса чувствительности с уровнем права доступа пользователя.

Данный алгоритм разработан на основе традиционных алгоритмов управления доступом пользователей к данным. Его достоинствами являются простота и небольшое количество операций, а также возможность обеспечения безопасности каждого триплета данных (как столбца в реляционных базах данных), что позволяет контролировать доступ пользователей к различным частям данных.

Контроль логических выводов

С семантическими данными пользователи могут использовать данные, имеющие низкий уровень безопасности, в логических операциях для получения данных, имеющих уровень безопасности больше уровня права доступа пользователя. Значит, пользователь нарушает правило безопасности информации в базе данных [5]. В данном разделе даются основные определения логических правил и видов графов-данных, описываются предлагаемые алгоритмы обнаружения и контроля логических выводов в семантической базе данных.

Основные логические правила

Если $P = \{A, B, C,\ldots, Z\}$ – это множество всех ресурсов (объект или субъект), а $X = \{X_1, X_2, ..., X_n\}$, где $n \geq 1$ – это множество всех бинарных отношений, тогда для выполнения логических выводов в семантической базе данных пользователи могут использовать следующие основные правила вывода:

- *симметричность*: если $X_i \in X$ является симметричным бинарным отношением, то для любых объектов $A, B \in P$ справедливо выражение $AX_iB \rightarrow BX_iA$;
- *транзитивность*: если $X_i \in X$ является транзитивным бинарным отношением, то для любых объектов $A, B, C \in P$ справедливо выражение $(AX_iB$ и $BX_iC) \rightarrow AX_iC$, где $A \neq B \neq C$;
- *импликация*: если бинарное отношение $X_i \in X$ заключает в себе отношение $X_j \in X$, то для любых объектов $A, B \in P$ справедливо выражение $AX_iB \rightarrow AX_jB$;
- *корреляция*: если $X_i \in X$ является корреляционным бинарным отношением, то для любых объектов $A, B, C, D \in P$ справедливо выражение $(AX_iB, CX_iD) \rightarrow AX_iC, A \neq C, B \neq C, A \neq D, B \neq D$;
- *декомпозиция*: если $X_i \in X$ является разложимым бинарным отношением, то для любых объектов $A, B \in P$ справедливо выражение $AX_iB \rightarrow AX_iA, BX_iB$, где $A \neq B$;

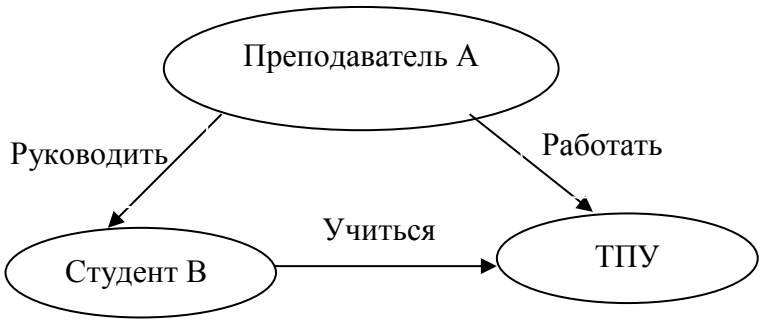

Рис. 2. Фрагмент RDF-графа семантической базы данных

Определение 1 (RDF-граф). Семантическая база данных представляет собой RDF-граф G, который состоит из множества вершин P (множество субъектов и объектов), множества ребер E (множество предикатов) и обозначается $G = (P, E)$, где каждая вершина является субъектом или объектом, каждое ребро является предикатом (отношение межу субъектом и объектом), существуют некоторые ребра между двумя вершинами [6]. На рис. 2 показан фрагмент RDF-графа семантической базы данных.

Определение 2 (видимый граф). Видимый граф для пользователя, имеющего уровень до-

ступа AC_s, является графом G_s, представляющим все триплеты, состоящие из троек: субъект, предикат, объект, к которым пользователь может иметь доступ, где $G_s \subseteq G$.

Определение 3 (логический граф). Логическим графом G^l_s является результат применения основных логических правил и добавления полученных результатов к графу G_s, где $G_s \subseteq G^l_s$.

Определение 4 (поток информации вершины графа). Потоком информации для вершины A является множество всех рёбер, непосредственно связанных с A. Данной поток обозначается как $C(A)$.

Правило (возможность выполнения логической операции между двумя вершинами). Предположим, что в графе G^l_s вершины A и B имеют потоки информации $C(A)$ и $C(B)$, где $C(A)$, $C(B) \subseteq G^l_s$. Если $C(A) \cap C(B) = \emptyset$, то пользователь не может получить логические выводы из этих известных ему вершин и их связей [4]. Из вышесказанного для определения возможности возникновения логических выводов из известных связей и вершин можно составить алгоритм (алгоритм 2), показанный на рис. 3.

В соответствии с данным алгоритмом, если между вершинами A и B обнаружены связи, то пользователь может применить логические правила и получать логические выводы из данных вершин. В противном случае, пользователь не может получить логических выводов из вершин A и B. Входными данными алгоритма 2 являются множества всех вершин и рёбер графа. Выходными данными являются вершины, между которыми имеется возможность выполнять логический вывод.

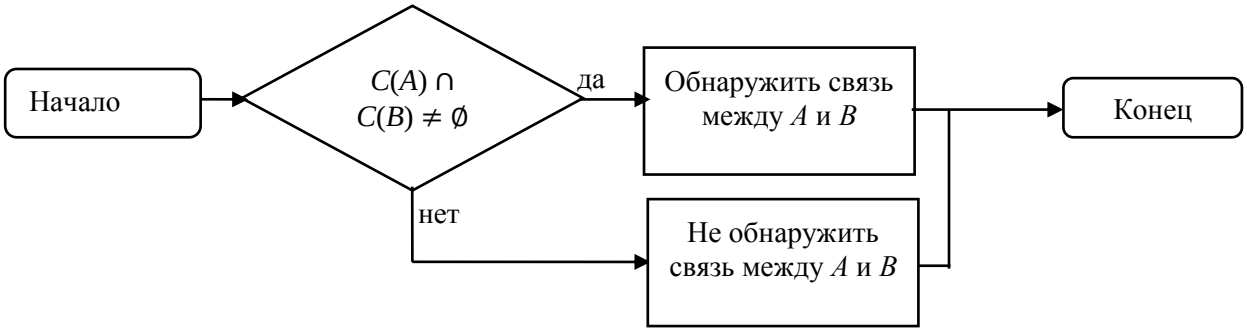

Рис. 1. Алгоритм определения возможности получения логических выводов из двух вершин

Предположим, что в графе G^l_s вершины A и B имеются потоки информации $C(A)$ и $C(B)$, где $C(A)$, $C(B) \subseteq G^l_s$. $C(A)$ имеет максимальный уровень m, $C(B)$ имеет максимальный уровень n, где $n \geq m$. Пользователь имеет уровень доступа $AC_s = cl$ и не имеет доступа к какой-то части данных. Для определения возможности получения логических выводов между вершинами A и B был построен алгоритм (алгоритм 3), показанный на рис. 4.

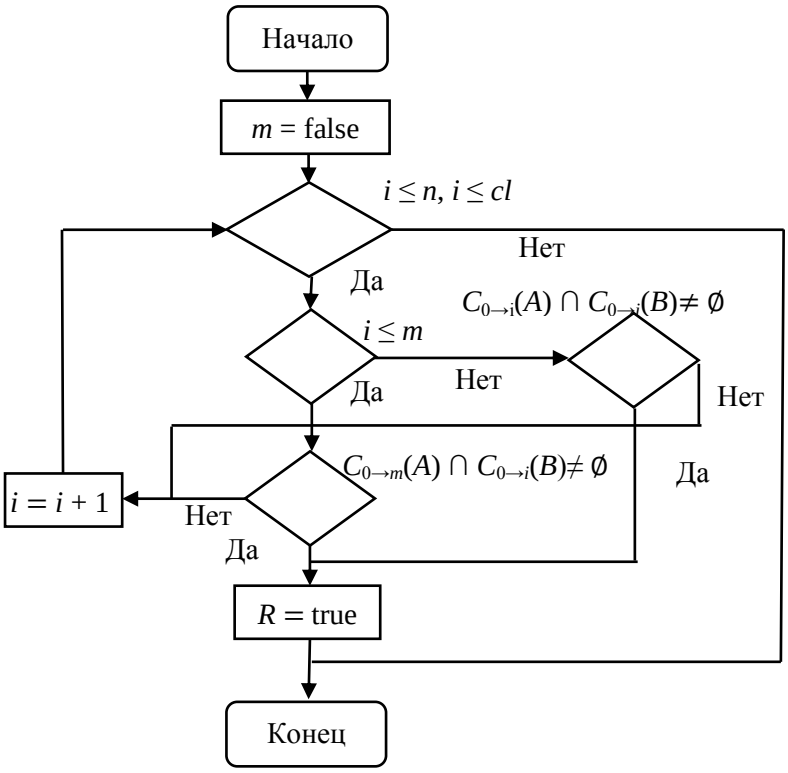

Рис. 2. Алгоритм определения возможности получения логических выводов из двух вершин, имеющих разные уровни права доступа

В соответствие с данным алгоритмом выполняются следующие основные операции:

- поиск общего потока информации между A и B на уровнях, которые меньшее, чем m. Если между ними обнаружены общий поток информации, то операция прекращается;
- поиск общего потока информации между A и B на уровнях, которые меньше, чем cl. Если между ними обнаружены общий поток информации, то операция прекращается.

В данном алгоритме используется переменная R для сохранения получаемого результата: если R = *true*, то можно получить связь между A и B; если R = *false*, то нельзя получить связь между A и B.

Несанкционированное получение логических выводов

Определение 5 (несанкционированный логический вывод). В семантической базе данных для любого пользователя s, имеющего уровень доступа AC_s, несанкционированным логическим выводом является процесс добавления дополнительного рёбра e к видимому графу G_s, где $e \in G \backslash G_s$.

Из этого определения следует, что вывод является нарушенным только в случае, когда полученная связь между вершинами находится в невидимой части данных.

Определение 6 (обнаружение нарушения логических выводов). Обнаружение нарушения логических выводов является процессом поиска всех связей, находящихся в невидимой части графа $G \backslash G_s$.

Для конкретного уровня права доступа пользователей после обнаружения выводов все связи будут разделены на два типа: раскрытые (могут являться несанкционированными логическими выводом) и безопасные (не могут являются несанкционированными логическими выводом).

Для обнаружения несанкционированных логических выводов был построен алгоритм (алгоритм 4), который включает следующие шаги:

Шаг 1: Применение алгоритма 1 для определения видимых частей графа пользователем: определение связей, у которых уровень меньше, чем уровень доступа пользователя; определение связей, у которых уровень больше, чем уровень доступа пользователя.

Шаг 2: Применение основных логических правил для получения логических графов G^l_s:

- если логические выводы находится в невидимых частях графа, то эти связи являются раскрытыми, и они будут отмечены;
- после этого поцесса остальные связи, имеющие высокой уровень безопасности (не находятся в невидимых частях графа), называются обычными связями.

Шаг 3: Применение вышесказанного правила для отмеченных вершин логического графа G^l_s. Если существует одна или несколько связей между двумя вершинами, то возможно применение логических правил.

- если невозможно получить вывод на более высоком уровне безопасности между двумя вершинами, то все связи между ними *являются безопасными;*
- если возможно получить вывод более высоком уровне безопасности между двумя вершинами, то все связи между ними *являются подозреваемыми;*

Шаг 4: Если существуют *подозреваемые связи*, то необходимо использовать теорию вероятностей для расчёта вероятности их выполнения. Такая возможность называется *вероятностным выводом*. Если вероятность вывода больше или равна *заданной вероятности p* (пороговое значение), то эти подозреваемые связи помечены как раскрытые связи, в противном случае они помечаются как безопасные связи.

На рис. 5 показан алгоритм обнаружения нарушения логических выводов. В данном алгоритме множество всех связей, имеющих более высокие уровни безопасности, чем уровень доступа пользователя AC_s, обозначено, как E_h, где $E_h = \{e_i: e_i \in G \backslash G_s$, где e_i является ребром$\}$, а множества всех раскрытых и безопасных связей, имеющих высокие уровня безопасности, обозначаются как E_d и E_{sa}.

С помощью данного алгоритма могут быть определены все безопасные и раскрытые связи. Пользователь не может получить логический вывод только в случае, когда все вершины, влияющие на данные связи, отмечены как невидимые.

Рис. 3. Алгоритм обнаружения нарушения логических выводов

Пример применения предложенного алгоритма для решения задачи

Рассмотрим следующий пример: пусть имеется семантическая база данных, которая содержит множество триплетов со своими индексами безопасности $R = \{R^1_{AB}(0100), R^5_A(0110), R^9_{AB}(1100), R^8_{AB}(0000), R^9_{AE}(1111), R^1_{BE}(0000), R^2_{BC}(1110), R^6_{CB}(0100), R^3_{CB}(1100), R^7_{DC}(0110), R^{10}_D(1100), R^4_{DE}(1110)\}$. Предположим, что пользователь имеет уровень доступа $AC_S = (1100)$.

Тогда требуется определить набор триплетов, которые пользователь может использовать для выполнения вывода данных, превышающих его уровень доступа AC_S.

На рис. 6 показан фрагмент RDF-графа. Используемые в данном графе отношения $\{X_1, X_2, \ldots, X_{10}\}$ обладают следующими свойствами: X_1 является симметричной связью, X_3 – транзитивной связью, $<X_6, X_8>$ – доминирующее правило, (в данной задаче $R^6_{BC} \rightarrow R^8_{BC}$). $<X_9, X_5>$ – доминирующее правило, т. е., если R_i и R_j доминирующее правило, то $xR_iy \rightarrow xR_jx$, $x \neq y$ (в данной задаче, $R^9_{AB} \rightarrow R^5_A$).

Для решения данной задачи безопасности семантической базы данных может быть применен алгоритм 4:

Шаг 1. определение видимых триплетов (граф G_s).

При выполнении прямой проверки прав доступа пользователям нельзя видеть триплеты $R^5_A(0110)$, $R^9_{AE}(1111)$, $R^4_{DE}(1110)$, $R^7_{DC}(0110)$, $R^2_{BC}(1110)$, так как их уровень доступа меньше, чем уровень безопасности триплетов. На рис. 7 показан видимый пользователю граф триплетов.

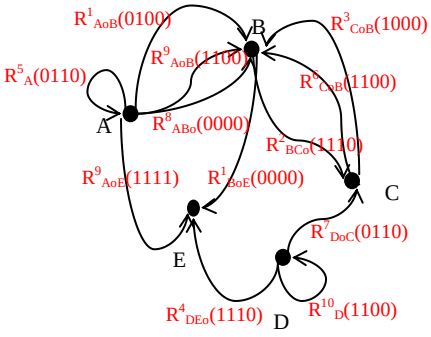

Рис. 4. Граф связи между элементами триплетов в базе данных

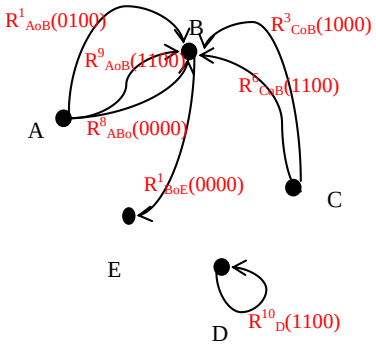

Рис. 5. Граф триплетов, видимый пользователем

Шаг 2. Определение логических и видимых триплетов (граф G^l_s).

В соответствие с заданными правилами R^9_{AB} доминирует над R^5_A и R^6_{CB} доминирует над R^8_{CB}, в результате получается логический граф, представленный на рис. 8.

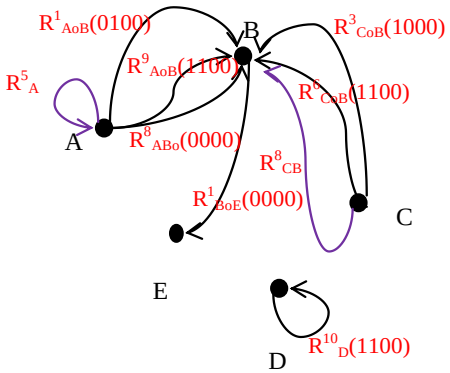

Рис. 6. Логический граф

Шаг 3. Определение раскрытых, безопасных и подозреваемых триплетов.

Один из двух логических результатов R^5_A является нарушенным триплетом, потому что $R^5_A \in G \backslash G_s$. В результате получаются: раскрытая связь R^5_A и неясные связи $R^9_{AE}(1111)$, $R^4_{DE}(1110)$, $R^7_{DC}(0110)$, $R^2_{BC}(1110)$. Для контроля неясных связей необходимо проверить такие пары вершин, как $\{<A, E>, <D, E>, <D, C>, <C, B>\}$.

Так как известно, что:

- $C(A) \cap C(E) \neq \emptyset$ (между вершинами A и E существует общая вершина B), значит, из них может следовать другой триплет;
- $C(D) \cap C(E) = \emptyset$, значит, из них не может следовать другой триплет;

- $C(D) \cap C(C) = \emptyset$, значит, из них не может следовать другой триплет;
- $C(B) \cap C(C) \neq \emptyset$, значит, из них может следовать другой триплет.

После этого шага получается множество нарушенных связей $\{R^5_A\}$, множество безопасных связей $\{R^4_{DE}(1110), R^7_{DC}(0110)\}$ и множество подозреваемых связей $\{R^9_{AE}(1111), R^2_{BC}(1110)\}$.

Шаг 4. Контроль наследованных триплетов.

Вначале проверяется связь $R^9_{AE}(1111)$ между вершинами A и E: основными триплетами, не влияющими на подозреваемую связь $R^9_{AE}(1111)$, являются $\{R^3_{CB}, R^6_{CB}, R^8_{CB}\}$; основными триплетами, влияющими на подозреваемую связь $R^9_{AE}(1111)$, являются $\{R^1_{AB}, R^9_{AB}, R^8_{AB}, R^8_{BE}\}$.

Тогда можно определить цепочки логического вывода $\{R^1_{AB}, R^1_{BE}\}$, $\{R^9_{AB}, R^1_{BE}\}$, $\{R^8_{AB}, R^1_{BE}\}$. С их помощью пользователь, с некоторой вероятностью, может выводить триплет R^9_{AE}.

Для того чтобы пользователь не смог вывести R^9_{AE}, в этих парах не должен присутствовать один из элементов. В данном случае пользователю нельзя видеть триплет R^1_{BoE} или триплеты $\{R^1_{AB}, R^9_{AB}, R^8_{AB}\}$.

Теперь переходим к проверки связи R^2_{BC} между вершинами B и C: связи R^1_{AB}, R^8_{AB}, R^9_{AB}, R^1_{BE} не имеют отношения с R^2_{BC}, а связи R^3_{CB}, R^6_{CB}, R^8_{CB} имеют отношения с R^2_{BC}. Из этих триплетов пользователь может вывести R^2_{BC} с некоторой вероятностью. Для того чтобы пользователь не мог это сделать, ему должны быть невидимы триплеты R^3_{CB}, R^6_{CB}, R^8_{CB}.

Выводы

Рассмотрены такие основные проблемы безопасности семантических баз данных, как контроль прямого доступа пользователей к данным и контроль логических выводов в семантических данных.

Для решения этих задач предложены алгоритмы контроля прямого доступа пользователей к триплетам RDF-данных и алгоритмы контроля логических выводов в семантических базах данных. Для контроля прямого доступа пользователей к базе данных была применена мандатная политика безопасности компьютерных систем. Для этого был разработаны общий метод для описания уровня безопасности триплетов RDF-данных и алгоритм для контроля доступа пользователей к RDF-данным. Достоинствами данного алгоритма являются простота и небольшое количество требуемых операции, а также возможность обеспечения безопасности каждого триплета данных, что позволяет контролировать доступ пользователей к различным частям данных.

Также были даны определения основным логическим правилам и понятиям некоторых видов графов RDF-данных для построения алгоритма контроля логических выводов в семантических данных. Данный алгоритм позволяет обнаружить и контролировать несанкционированные логические выводы.

СПИСОК ЛИТЕРАТУРЫ

10. Thuraisingham B. Secure Sematic web Services. – Texas: Department of Computer Science, 2007. – 123 p.
11. Anton B. Resource Description Framework // Technical report, Department of Mathematics and Computer Science, 2006. – V. 20 – № 3. – P. 157–168.
12. Девянин П.Н. Модели безопасности компьютерных систем. – М.: Академия, 2005. – 144 с.
13. Девянин П.Н. Анализ безопасности управления доступом и информационными потоками в компьютерных системах. – М.: Радио и связь, 2006. – 176 с.
14. Thuraisingham B. Security for the semantic web // International Journal of Approximate Reasoning. – 2005. – V. 11 – № 1. – P. 257–268.
15. Thuraisingham B. Database and Applications Security: Integrating Data Management and Information Security. – Texas: Department of Computer Science, 2005. – 276 p.

Поступила 10.09.2012 г.